

Document Generated: 11/16/2024

Learning Style: Virtual Classroom

Provider: Cisco

Difficulty: Intermediate

Course Duration: 5 Days

Network Programmability for the Data Center (NPDC)



About this course:

Network Programmability for the Data Center (NPDC) v1.0 is a 5 day instructor led course. This course teaches how to automate an end-to-end Cisco Data Center consisting of Nexus 7000 and 9000 switches. This course also includes Cisco ACI

and UCS platforms using state of the art Model Driven API's. Applying CCIE-style learning methods, the NPDC course places emphasis on learning via hands-on labs and hands-on demonstrations. The terms “proof-by-Python” and “proof-by-JSON or XML” will be used frequently throughout the Data Center training course. It is recommended that students who take this course have the knowledge and skills of a CCNP professional, in addition to having completed the Programming for Network Engineers (PRNE) ELT or equivalent Python programming experience. This NPDC training is intended for Data Center engineers who wish to learn how to deploy network programmability and automation using various Cisco platforms listed below.

Course Objective:

Upon completing this course, the learner will be able to meet these overall objectives:

- Use Linux tools that empower network programmers
- Write and troubleshoot Python scripts in the specialized area of Network Programmability
- Understand and interact with Data Models (both the Cisco Managed Object Model and YANG models) on Nexus switches, the ACI APIC and the UCS
- Understand and interact with REST, NETCONF and RESTCONF API's on Nexus switches, the ACI APIC and the UCS
- Understand and interact with XML and JSON on Nexus switches, the ACI APIC and the UCS
- Learn how to use DevOps tools such as GitHub, VIRT and services offered by Cisco DevNet
- Survey the capabilities of OpenStack and Cisco CloudCenter

Audience:

The primary audience for this course is as follows:

- This course is designed for Data Center engineers who want to learn how to deploy network programmability and automation using the following Cisco platforms: the NEXUS 7000, the NEXUS 9000 in both NXOS-mode and ACI mode, the ACI APIC and the UCS.

Prerequisite:

The knowledge and skills that a learner should have before attending this course are as follows:

- CCNP or Equivalent Experience
- Complete the Programming for Network Engineers (PRNE) ELT or equivalent Python programming experience

Course Outline:

Module 1: Network Programmability Fundamentals

- Lesson 1: Introduction to Network Programmability
- Lesson 2: Understanding Software-Defined Networking
- Lesson 3: Explore Network Programmability and Automation
- Lesson 4: Examine Open Source Tools and Enterprise Platforms
- Lesson 5: Understand Network Programmability Technology
- Lesson 6: Explore a range of Cisco Platforms and their APIs

Module 2: Linux Primer for Network Engineers

- Lesson 1: Understand the Need to Learn Linux
- Lesson 2: Explore the Linux File System
- Lesson 3: Examine Linux Distributions and Package Managers
- Lesson 4: Work with Files and Directories
- Lesson 5: Explore Linux Processes

Module 3: Linux Tools for Programmers

- Lesson 1: Build a Programmer's Workbench in Linux
- Lesson 2: Use Complex Grep and Find Commands to Locate Files
- Lesson 3: Explore powerful use cases of BASH scripting for network programmers

Module 4: Python Fundamentals for Network Engineers – Part 1

- Lesson 1: Understanding Python
- Lesson 2: Executing Python Code
- Lesson 3: Examining Python Helper Utilities and Function
- Lesson 4: Writing Idiomatic Python
- Lesson 5: Exploring Common Python Data Types

Module 5: Python Fundamentals for Network Engineers – Part 2

- Lesson 1: Lists
- Lesson 2: Dictionaries
- Lesson 3: Loops
- Lesson 4: Functions
- Lesson 5: Working with Files

Module 6: Writing and Troubleshooting Python Scripts

- Lesson 1: Writing Scripts
- Lesson 2: Executing Scripts
- Lesson 3: Analyzing Code
- Lesson 4: Error Handling

Module 7: Python Libraries

- Lesson 1: Python Libraries

- Lesson 2: Python Modules
- Lesson 3: Python Packages

Module 8: Data Models in the Data Center

- Lesson 1: Introduction to Data Models
- Lesson 2: Data Models Defined
- Lesson 3: The power of using a Data Model
- Lesson 4: The pain of not using a Data Model
- Lesson 5: Review of the SNMP Data Model
- Lesson 6: SNMP compared to the Cisco Data Center Managed Object Model
- Lesson 7: SNMP compared to a YANG derived Data Model
- Lesson 8: The Power of YANG + Cisco DevNet ydk-gen in auto-generating Python Modules
- Lesson 9: An Example of a State of the Art Network Programmability environment using Python with a Data Model

Module 9: YANG Data Modeling

- Lesson 1: YANG Overview
- Lesson 2: YANG Module and Module Header
- Lesson 3: YANG: Fundamental Definitions and Statements
- Lesson 4: YANG Types and the TypeDef Statement
- Lesson 5: YANG Choice and Grouping Statements
- Lesson 6: YANG Miscellaneous Statements
- Lesson 7: Putting Things Together with YANG
- Lesson 8: YANG Model Examples with XML and JSON

Module 10: YANG Tools

- Lesson 1: YANG Validator
- Lesson 2: The YANG Tool
- Lesson 3: YANG Development Kit
- Lesson 4: YDK-Py API Structure
- Lesson 5: YDK-Gen
- Lesson 6: YANG Explorer

Module 11: Network Programmability on the NEXUS 7000 and 9000

- Lesson 1: Introduction to Network APIs and Protocols
- Lesson 2: Evolution of Device Management and Programmability
- Lesson 3: Data Encoding Formats
- Lesson 4: JSON
- Lesson 5: XML
- Lesson 6: Model-Driven Programmability Stack
- Lesson 7: REST
- Lesson 8: NETCONF
- Lesson 9: RESTCONF
- Lesson 10: gRPC

Module 12: NEXUS 7000 Programmability

- Lesson 1: Nexus Programmability Overview
- Lesson 2: NX-API CLI—Part 1
- Lesson 3: NX-API CLI—Part 2
- Lesson 4: NETCONF
- Lesson 5: Python on Box
- Lesson 6: NX-API REST

Module 13: Nexus 9000 Programmability

- Lesson 1: Review of the many programmability options of the NEXUS 9000
- Lesson 2: Programming the NEXUS 9000 using REST and the Cisco Managed Data Model
- Lesson 3: Browsing the NEXUS 9000 Managed Data Model using Visore
- Lesson 4: Using the NXOS CLI to verify, debug and log network programmability on the NEXUS 9000
- Lesson 5: Exploring the on-board Python programmability using the NEXUS 9000 CLI, BASH and GUESTSHELL interfaces
- Lesson 6: Configuring NETCONF and RESTCONF with YANG Models on the NEXUS 9000

Module 14: Programming the ACI APIC

- Lesson 1: Cisco Application Centric Infrastructure
- Lesson 2: ACI Overview
- Lesson 3: ACI Object Model
- Lesson 4: APIC REST API
- Lesson 5: Using Postman REST Client
- Lesson 6: Cobra SDK
- Lesson 7: Arya
- Lesson 8: ACI Toolkit
- Lesson 9: ACI Toolkit Applications — CLI
- Lesson 10: ACI Toolkit Applications — Diagrams Tool
- Lesson 11: ACI Toolkit Applications — Lint
- Lesson 12: ACI Toolkit Applications — Cable Plan
- Lesson 13: ACI Toolkit Applications — Event Feeds
- Lesson 14: ACI Toolkit Applications — Fake APIC

Module 15: Programming the Cisco UCS

- Lesson 1: UCS Programmability
- Lesson 2: Understand the Basic Physical UCS Components as a Single System
- Lesson 3: Explore XML and via the UCS Manager
- Lesson 4: Explore the XML content of a backup configuration script generated by the UCS Manager
- Lesson 5: Explore the Cisco Managed Object Model of the UCS and compare its similarities to the Managed Object Models found on the NEXUS 9000 and the ACI APIC.

- Lesson 6: Write Python scripts using both the UCS Data Model and the UCS Manager SDK from DevNet
- Lesson 7: Use the exact same programmability tool kit with UCS Central to scale to a multi-UCS domain environment
- Lesson 8: Explore UCS Director Orchestration and Workflow creation. Compare its focus on JavaScript to the use of Python with UCS Manager and UCS Central.
- Lesson 9: Formulate a learning roadmap to master UCS Programmability featuring the UCS Platform Emulator

Module 16: Data Center DevOps

- Lesson 1: Version Control
- Lesson 2: Version Control Systems
- Lesson 3: Overview of Git
- Lesson 4: Git Architecture
- Lesson 5: Git Commands
- Lesson 6: Git Workflow
- Lesson 7: Git Branches
- Lesson 8: Using Git
- Lesson 9: Collaborating with GitHub
- Lesson 10: GitHub Pull Request: Fork and Pull
- Lesson 11: Changing Views

Module 17: Automated Testing

- Lesson 1: Network Test Infrastructure
- Lesson 2: Network Function Virtualization
- Lesson 3: VIRT
- Lesson 4: DevNet
- Lesson 5: DevNet Sandbox
- Lesson 6: DevNet Learning Labs
- Lesson 7: DevNet GitHub
- Lesson 8: Network Testing
- Lesson 9: Unit Tests

Module 18: OpenStack and Programmability

- Lesson 1: Need for Programming
- Lesson 2: The Business Need for Network Programmability
- Lesson 3: ACI Programmability
- Lesson 4: ACI Open APIs & Ecosystem
- Lesson 5: API Protocols

Module 19: CloudCenter Programmability

- Lesson 1: CloudCenter API Overview
- Lesson 2: CloudCenter Integration Fundamentals - UCS Director
- Lesson 3: Cisco Cloud Center and ACI

Credly Badge:



Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)